AD-A112 168 NAVAL OCEAN SYSTEMS CENTER SAN DIEGO CA F/G 9/2

INTEGRATION OF NARRATIVE PROCESSING, DATA FUSION, AND DATABASE --ETC(U)

OCT 81 R A DILLARD

UNCLASSIFIED NOSC/TD-480 NL

1 OF 1

AD A
112168

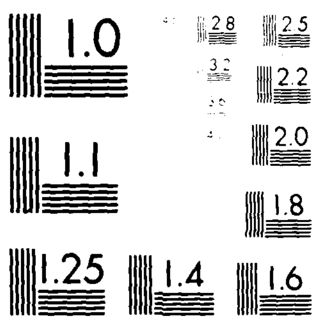
NOSC

END

DATE FILMED

04-82

DTIC



MICROCOPY RESOLUTION TEST CHART

ADA112168

(12)

NOSC

NOSC TD 480

NOSC TD 480

Technical Document 480

INTEGRATION OF NARRATIVE PROCESSING, DATA FUSION, AND DATABASE UPDATING TECHNIQUES IN AN AUTOMATED SYSTEM

Robin A. Dillard

29 October 1981

Prepared for
Naval Electronic Systems Command

Approved for public release; distribution unlimited

NAVAL OCEAN SYSTEMS CENTER
SAN DIEGO, CALIFORNIA 92152

NAVAL OCEAN SYSTEMS CENTER, SAN DIEGO, CA 92152

AN ACTIVITY OF THE NAVAL MATERIAL COMMAND

SL GUILLE, CAPT, USN
Commander

HL BLOOD
Technical Director

ADMINISTRATIVE INFORMATION

The work covered in this document was done during fiscal year 1981 for the Naval Electronic Systems Command, Code 613.

Released by
R. C. Kolb, Head
Tactical Command and
Control Division

Under authority of
J. H. Maynard, Head
Command Control –
Electronic Warfare Systems
and Technology Department

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER Technical Document 480 (TD 480)	2. GOVT ACCESSION NO. AD-A112 168	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) INTEGRATION OF NARRATIVE PROCESSING, DATA FUSION, AND DATABASE UPDATING TECHNIQUES IN AN AUTOMATED SYSTEM		5. TYPE OF REPORT & PERIOD COVERED FY 1981 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Robin A. Dillard		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Ocean Systems Center San Diego, CA 92152		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61153N, XR01408 RX0140801, 824-CC25
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Electronic Systems Command Washington, DC 20360		12. REPORT DATE 29 October 1981 13. NUMBER OF PAGES 60
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Artificial intelligence
Production systems
Natural Language Processing
Data fusion

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Narrative data processing and database updating are examined from the viewpoint of interaction with other processes in an automated data fusion system. Methods are outlined for using production rules (if-then rules) in the processing of the narrative data and for integrating such a process with other rule-based data-fusion processes. Database updating methods which employ a "history file" are processed. An experimental rule-based program compares event reports for redundancies, combines reports of the same event and retires old data into a history file in a form useful for event reconstruction and evaluation. A number of related areas of investigation are suggested for future research.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-LF-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

S/N 0102-LF-014-6601

CONTENTS

FIGURES

ACRONYMS

NLP	Natural Language Processing
ONSTA	On Station
PTAPS	Platform-Track Association Production Subsystem
Rosie	Rule-oriented system for implementing expertise
STAMMER	System for Tactical Assessment of Multisource Messages, Even Radar
TOI	Target of Interest
TSA	Tactical Situation Assessment

1. INTRODUCTION

This report discusses two phases of investigations into the integration of data fusion techniques. One is the continuing investigations of the computer understanding of narrative data and the use of the processed data in an automated data fusion system. The discussion here concerns the feasibility of using production rules (if-then rules) in the processing of the narrative and the integrating of such a process with other rule-based data-fusion processes. The narrative data of particular interest are the narrative comments in RAINFORM messages, ASW (antisubmarine warfare) messages which can include natural language comments that amplify the formatted information. The other phase of investigations concerns the updating of the database of the data fusion system. In connection with these investigations, the concept of a "history file" is introduced.

1.1 Data Fusion System Model

This effort is part of a larger effort to develop automated data fusion techniques [1]. Under the larger effort, a conceptual model of an automated data fusion system (Figure 1-1) has evolved. The goal has been to identify the voids in the technologies required to support the various functions of the system and to devise techniques to fill these voids. Complementary interaction of individual fusion techniques has been the underlying design philosophy in the concept formulation of this integrated system.

The only component of the "final fusion" box of Figure 1-1 in experiments and investigations to date has been a "production system," a system which implements a number of if-then rules called "productions." (Systems of three different structures have been employed in the experiments.) In addition to various control and support mechanisms, a production system must have a database. However, this database is shown separately in Figure 1-1 because ideally it would be shared with other processes. (The components of the database are discussed in Section 1.2.) In particular, it is important that the natural language processing (NLP) functions and the database updating functions have access to the database, a need which is addressed in this report. Furthermore, it is envisioned at this point that the updating functions and some of the NLP functions (represented by the "NLP Control" box) in Figure 1-1 should be performed within the same production system. Sections 3 and 4 discuss rule-based approaches to NLP and database updating, after a summary of present experimental NLP programs in Section 2.

```
                                     ----------
                                     |  NLP   |
                   * * * * * * * * CONTROL* * * *
TEXTUAL           -------*---------  |        |     *
----------------->| NLP* AND DATA  | ----------     *
INFORMATION       |  STRUCTURING   | ----------  ---*--------------
                  |  - - - - - -   |->|        |  |      | PRIMARY |
----------------->| SPECIAL NLP AND| |        |  | DATA-| MEMORY  |
PARTLY FORMATTED  |DATA STRUCTURING| |DATABASE|  | BASE |         |
MESSAGES          ------------------ |UPDATING|  |INTER-|---------|
                                     |        |->| FACE |SECONDARY|
                  ------------------ |        |  |      | MEMORY  |
----------------->|DATA STRUCTURING|->|        |<-|      |         |
FULLY FORMATTED   |                | |        |  |      |---------|
MESSAGES          ------------------ |        |  |      | HISTORY |
                                     |        |  |      |  FILE   |
------------------------------------>|        |  |      |         |
SLOW PERISHING DATA UPDATES          ----------  -----------------
                                         ^          *
                                         |          *
                                     ----------------
                                     |              |  USER
                                     | FINAL FUSION |-> RETRIEVAL
                                     |              | AND ALERTING
                                     ----------------     SYSTEMS
```

*Natural Language Processing

Figure 1-1: Model of an integrated data fusion system. The arrows indicate data flow and the asterisks indicate other interaction, particularly access to data.

One technique developed under the larger effort is PTAPS, a Platform-Track Association Production Subsystem [2], [3], [4]. PTAPS is a method of extending the capabilities of a production system applied to tactical data fusion, to enable the system to perform the higher order logical reasoning required to solve certain kinds of platform identification problems. PTAPS would exist as a subsection of the "Final Fusion" box of Figure 1-1 .

1.2 Database

The database subsystem of the system shown in Figure 1-1 contains a "primary memory," a "secondary memory" and a "history file" and these are explained below. Additional domain knowledge could be contained in the NLP subsystems. Private or temporary memories also are needed at most levels of manipulation and computation.

Primary Memory: Primary memory is the storage of (a) all domain knowledge data needed by the production system, (b) input message data and (c) data created by the firing of production rules -- with the exception of infrequently needed domain knowledge and old, perishable data. This memory could be partitioned into multiple memories, each activated and deactivated as necessary.

Secondary Memory: Secondary memory is the storage of domain knowledge infrequently needed by NLP and fusion processes.

History File: History file is the storage of event data, as "understood" by the NLP subsystems and the final fusion subsystem.

A "history file" is needed to record events of tactical exercises and engagements in a manner useful in event reconstruction and evaluation. Good records are needed for assessment of fleet performance and readiness. Also, in determining the most probable enemy reaction to an operation under consideration, historical records of the enemy's strategies and reactions during earlier operations of a similar kind can be a valuable asset. The records can also be useful in estimating the probable outcome and losses of an operation and in deriving enemy capabilities.

All of the available data needed for good records will at some time be stored in the database of the tactical data fusion system envisioned for the future and now modelled on a small scale. Sophisticated knowledge-based system techniques are needed for selecting pertinent data, reconstructing events from them, and organizing the fused event records file for querying by other systems and humans. Here we consider the history file only from the aspect of entering into it partially fused event records from the data fusion system, which allows the data to be gracefully forgotten by the primary memory, but retrievable from the history file.

2. NLP OF PARTLY FORMATTED MESSAGES

2.1 NLP Investigations

Efforts to develop NLP techniques for processing Navy tactical narrative began with the study of a large collection of messages to determine the contextual problems, vocabulary statistics and syntactic characteristics of the natural language data. Reference [5] addresses these problems in terms of relevant NLP techniques, describing the techniques and giving sample "knowledge representations" of some of the narrative data.

An unclassified set of models of RAINFORM messages was constructed for NLP experiments. While the formats used in these messages were invented for this effort and the names, locations and times given in the actual messages were changed, the message models still exhibit the semantic and syntactic problems to be encountered by a message-understanding system. Several examples are given below. The first "word" of each line is the line name, or "identifier," and dictates the contents of the succeeding slots of the line. For example, the first line of each message is defined by

PRELIM/sender/month and day/msg #/msg type

where the message number starts at "1" each day for each sender. The various lines and the order in which they can occur in the message models are described in Reference [6].

Sample Message Models

```
PRELIM/TU 34.5.4/MAR 19/2/SIGHTING
MILAIR/BOMBER/BADGER/HIGH/40/UR
POSIT/33.25N/10.09W
POSAMP/192100Z3/VISUAL/GAINED
POSIT/33.25N/10.09W
POSAMP/192105Z8/VISUAL/LOST/300K/170T
NARR/VISUALLY SIGHTED BADGER AIRCRAFT APPROACH FROM SOUTHEAST
HEADING 350 DEGREES T. OVERFLEW BRUMBY, TURNED TO COURSE
150 DEGREES T, OVERFLEW BRUMBY AGAIN AND WENT INTO A CIRCULAR
PATTERN AT 10NM BEARING 170 DEGREES T FOR A FEW MINUTES.
AIRCRAFT THEN DEPARTED AREA ON HEADING 170 DEGREES T.
```

STOP

- - - - -

PRELIM/TG 21.2/MAR 20/2/SIGHTING
SURFMIL/AGI//UR/KRYM
POSIT/34.13N/8.31W
POSAMP/200335Z3/VISUAL/LOST/15K/250T
POSIT/34.17N/8.38W
POSAMP/200350Z0/VISUAL/REGAIN/25K/205T
POSIT/34.16N/8.38W
POSAMP/200400Z6/VISUAL/HOLDING/25K/205T
POSIT/34.06N/8.48W
POSAMP/200445Z5/VISUAL/LOST/13K/247T
LAMP/LOST VISUAL DUE DARKNESS
NARR/AGI KRYM MANEUVERED VICINITY OF CONSTELLATION DURING VERTREP/UNREP AT RANGE 1.5 TO 2.5NM BETWEEN 90 TO 230 RELATIVE BEARING. NO HARASSING TACTICS.
STOP

- - - - -

PRELIM/CGN 39/MAR 23/8/SIGHTING
SUBSURF/HIGH/TX-5/UR///DIESEL
POSIT/32.39N/13.12W
POSAMP/230105Z1/VISUAL/GAINED
NARR/SIGHTED GREEN FLARES THEN PERISCOPE AND ANTENNAS BEARING 192 APPROX 5 MILES FROM THIS UNIT WHILE REFUELING ALONGSIDE USS FLINT. SUBMARINE WENT SINKER. NO ATTACKS MADE. PERISCOPES AND ANTENNAS INDICATE DIESEL.
STOP

- - - - -

PRELIM/TU 34.7.0/MAR 21/3/MISSION SYNOPSIS
PRIOR/TU 34.7.0/MAR 21/2
GOAL/PREPLAN/ANTISUB
TOUR/SQUAD-20/2/210312Z9/210350Z1/210715Z6/210724Z6
AREA/34.00N/11.03W/25NM
ACFT/S-3A/#
CREW/....
NARR/CREW CONDUCTED ACOUSTIC, RADAR AND VISUAL SEARCH FOR DELTA APPROXIMATELY 125 MILES SOUTHEAST OF CARRIER. NO CONTACT GAINED. AT 210615Z5 AX DIRECTED CREW TO LAY BUOYS 3 MILES AFT OF CARRIER. NO CONTACT GAINED
STOP

- - - - -

PRELIM/TU 175.7/MAR 25/46/MISSION SYNOPSIS

PRIOR/TU 175.7/MAR 25/44
GOAL/PREPLAN/ANTISUB
TOUR/SQUAD-20/6/252007Z6/252020Z1/252315Z8/252348Z4
SUBSURF/HIGH/DN-26/UR/DELTA/HIGH/NUCLEAR
POSIT/34.25N/8.25W
POSAMP/252113Z4/VISUAL/GAINED/12K/150T
SONO/34.25N/8.24W/252120Z2
ACFT/S-3A/#
CREW/...
NARR/DELTA WAS ON SURFACE 2NM FROM CARRIER. CREW SIGHTED SUB ON THE SURFACE TRACKING 150T/12KTS AND DROPPED BUOY PATTERN AROUND SAME. SUB PROMPTLY SUBMERGED. TRACKED SUB FOR 1 HR 30 MIN THEN LOST CONTACT. DEPARTED STATION AND RETURNED TO CARRIER.
STOP

\- - - - -

Although not evident in these sample messages, most of the narrative information concerns routine operations or observations of minor importance. Furthermore, in the application of immediate concern here, the narrative information is not useful unless pertinent to the conditions of the production rules in the production system performing tactical data fusion. Processing all of the narrative lines can therefore be wasteful of computer resources. For this reason, the approach taken in this work has been to search the narrative lines for an indication of any of a number of relevant situations and, if there is an indication, to perform a follow-up analysis, aborting it whenever a determination is made that the situation discussed is not the one suspected.

An experimental NLP program based on this top-down approach, using keyward patterns as indicators of situations, was written in Interlisp and tested on a number of unclassified message models [6]. Narrative comments concerning certain overflight and geometry-reference situations pertinent to production system rules were successfully understood and were restructured into assertional statements read and used by STAMMER2 [7], a production system performing tactical situation assessment (TSA). The initial experimental results helped to demonstrate the feasibility of the technical approach. Since then, a new NLP program has been written [8] which has improved syntactic analysis portions and which can be extended easily to a large number of situations. The interface of these NLP programs with STAMMER2 is described in the next section.

2.2 Original Interfacing of NLP and STAMMER2

Before actual NLP begins, preprocessing functions convert the incoming message into a form suitable for further manipulation in Interlisp and interpret the information in the formatted lines. The sentences in the narrative parts are processed in specialized ways, and the "actor" or "object" is determined, where necessary, from formatted data or earlier narrative. The two kinds of situations recognized by the first NLP programs are an overflight of a friendly ship by a hostile aircraft and a referencing of the position of one ship with respect to another (e.g., "came within" and "in vicinity of"). In the sense that the analysis is aimed at identifying certain properties of events, the process is similar to filling slots in a frame [9] or script [10], and also bears resemblance to the "event template" technique [11] designed for processing the narrative portions of Air Force intelligence messages.

The production system chosen for initial experiments was STAMMER2, a development of NOSC [7]. The original STAMMER [12], A System for Tactical Situation Assessment of Multisource Messages, Even Radar, was developed to serve as a demonstration of the applicability of rule-based inference techniques to the problem of tactical situation assessment (TSA) and STAMMER2 is an improved version.

STAMMER2 accepts data in the form of two-node assertions, generally written as a triple (<relation> <node-a> <node-b>) and read "<node-b> is a <relation> of <node-a>" or "the <relation> of <node-a> is <node-b>." The assertion (SOURCE MESSAGE92 ENTERPRISE), for example, expresses the fact that the source of message 92 was the Enterprise. The relation "is" in STAMMER2 indicates that <node-b> "is a" <node-a>, and is expressed in assertional statement form as (<node-b> <node-a>). STAMMER2 also accepts and uses assertions containing more than two nodes, but these are currently used only internally and not for input data.

Computational functions called "oracles" provide an escape to LISP code for certain relations that are either difficult or tedious to represent as rules, and for relations that should only be calculated on demand ([7], Section 2.5 of Volume 1). The oracles available in STAMMER2 are listed in Volume 2 of Reference [7]. Oracles can be used like assertions in the conditions of STAMMER2 rules. Among the simplest two-argument oracles that return either T or nil are SAME-AS, ROUGHLY-THE-SAME-AS, GREATER-THAN and LESS-THAN. Oracles that return an answer ("lastarg" oracles, where the last "argument" is returned as an answer) include SPEED, RANGE and COURSE.

Figure 2-1 illustrates the major functions and components of STAMMER2 and of the NLP system as structured in the original program. The relationships among the various components of STAMMER2 are too complex to describe here but are indicated with asterisks; the data flow is illustrated with arrows. The narrative processing generates temporary data kept as property lists and variable bindings to word lists, numbers, etc., and STAMMER2 generates similar temporary data; consider these data to be distributed among the support boxes.

Although in practice all pertinent formatted data would be made available to STAMMER2, in this experimental version only those data from messages containing pertinent narrative are made available, via a message file. The message file is filled with messages of the form (MESSAGE <assertion 1> <assertion 2> ... <assertion n>).

```
                        NLP SYSTEM                  STAMMER2
                            ^                          ^
                   __________ __________      __________ __________
                  /                     \    /                     \

                    -----------------      ---------
                    | RESTRUCTURING |      |MESSAGE|
                    | OF DATA FROM  |--->|RECEIPT|
                    | FORMATTED AND |      ---------
                    |NARRATIVE LINES|      |
                    -----------------      |  -----------
                       ^           ^       -->|ASSERTION|
                       |           |          | DATABASE|****
                       |           |       -->|         |   *
                       |           |       |  -----------   *
                       |           |       |                *
                 ---------------   |       -------------    *
                 |             |   |       |RULE       |******
                 |   NLP OF    |   |       |INTERPRETER|    *
                 |  NARRATIVE  |<--|       -------------    *
                 |    LINES    |   |                        *
                 |             |   |         ----------     *
                 ---------------   |         |TSA RULES|******
                       ^           |         ----------     *
                       |           |                        *
                       |  FORMATTED|DATA     -------------  *
             NARRATIVE |  PERTINENT|TO       |TSA ORACLES|****
                 LINES |      NLP &| TSA     -------------  *
                       |           |                        *
                     ---------------         -------------  *
PARTLY FORMATTED     |   INITIAL   |         | OTHER     |  *
-------------------->|RESTRUCTURING|         | COMPUTING |  *
     MESSAGES        |& EXTRACTION |         |  SUPPORT  |****
                     ---------------         |(GEOMETRY, |  *
                                             |CONFIDENCE,|  *
                                             |     ETC.) |  *
                                             -------------  *
                                                            *
                                          ----------------  *
                                          | USER SUPPORT |  *
                                          |(EXPLANATION, |***
                                          |GRAPHICS, ETC.)|
                                          ----------------
```

Figure 2-1: Original NLP system and its interaction with STAMMER2.

3. RULE-BASED NLP

3.1 Introduction

Much of the reasoning involved in processing the narrative lines can be expressed in the form of production rules. These can be, for example, pattern-action rules used in parsing, rules that use the formatted lines in understanding the narrative (e.g., if there is a ship S reported in the narrative and there is no formatted line consistent with S, then S is not hostile), and rules that infer activities (e.g., if the primary function of a hostile ship is reconnaissance and it remains in the area of a US ship, then its activity is probably reconnaissance). An examination of the following message is useful in demonstrating the difficulty of formulating good rules.

```
PRELIM/TG 34.2/MAR 20/1/MISSION SYNOPSIS
PRIOR/TG 34.2/MAR 19/6
GOAL/TRAINING/ANTISUB
DURATION/HOLDING/191747Z9/192125Z0/PASV ACOUS
TOUR/SQUAD 47/12/191358Z7/191720Z0/200155Z3/200355Z5
ELLIP/34.32N/11.00W/360T/150NM/130NM
SUBSUM/VISUAL/ACOUS/ACOUS/NC/191720Z0
SUBSURF/CERTAIN/NL-11////NUCLEAR
POSIT/34.41N/10.07W
POSAMP/191912Z3/VISUAL/FIXED/7K/055T
SONO/34.28N/10.23W/191911Z2
SONO/34.28N/10.23W/192304Z9
LAMP/NO CONTACT
SONO/33.18N/11.05W/200013Z6
LAMP/NO CONTACT
DOWN/RADAR/191435Z3
LAMP/RADAR DOWN RESTRICTED SEARCH FOR AGI KRYM
ACFT/P-3/#
CREW/...
NARR/TASKED TO RELOCATE AGI KRYM PRIOR TO ASW OPERATIONS.
RADAR DOWN, SEARCHED VISUALLY 45NM AREA AROUND 34.00N 9.00W
WITH NO CONTACT.  PROCEEDED TO ONSTA AND JOINED ASW SHIPS.
RECEIVED HOT CONTACT ON TOI FROM ONSTA AIRCRAFT.  MAINTAINED
CONTACT FROM 191747Z9 TO 192125Z0.  SUB WAS REMAINING IN
RELATIVELY SMALL AREA AND PINGED SONAR PRIOR TO SURFACING.
LOST CONTACT 192125Z0.  DEPARTED AREA.
STOP
```

Consider the last sentence of the narrative -- "DEPARTED

AREA." The reader knows that the subject is the "crew" and that the area is not specifically the "45NM AREA" but some general area related to "ONSTA" (on-station) and the "RELATIVELY SMALL AREA." Depending on the extent of the TSA rules, the system may need to be capable of rejecting the interpretation "the sub departed the relatively small area." A useful rule for this sentence would be one that takes the subject of the previous sentence, "LOST CONTACT 191252Z0." If this rule had been applied to that previous sentence, though, the subject for both sentences would be "sub" instead of "crew." A very good rule for this message, but not necessarily for others, is one that suggests the subject is "crew" if none is given. Other possible subjects, taken from earlier narrative, are: ONSTA AIRCRAFT, TOI, ASW SHIP and AGI KRYM. The system may previously have concluded that the "TOI" (target of interest) is the sub and not the AGI, partly from the fact that no formatted line was given for the AGI. Other problems occur with other sentences. For example, the segment "PINGED SONAR PRIOR TO SURFACING" tells the reader that the visual sighting resulted from a surfacing of the sub, not just a sighting of a periscope or snorkel. The system should easily eliminate other platforms as the thing "SURFACING" by noting that the sub is the only one capable of surfacing.

One reasonable approach to dividing these kinds of ambiguity problems between an initial NLP unit and a production system which also deals with the larger picture of TSA is to have the initial NLP unit present to the production system, with each ambiguous conceptual structure, a list of candidate actions, actors, objects or whatever, and their respective initial probabilities or weights. The production system would use the formatted data and additional knowledge (as derived from prior sentences and messages) about the situation to select the most probable. The certainty of the decision would have to be high unless data certainty factors are carried in the TSA system.

A complete understanding of the information in message narrative requires a considerable amount of causal inferencing. For example, the narrative in the last message model in Section 2.1 includes the lines:

CREW SIGHTED SUB ON THE SURFACE TRACKING 150T/12KTS
AND DROPPED BUOY PATTERN AROUND SAME. SUB PROMPTLY
SUBMERGED.

The reader immediately realizes that the sighting of the sub was a contributing cause of the dropping of a buoy pattern and

that the crew's presence was a contributing cause of the sub's submerging and also that tactical and self-preservation considerations caused the sub to have as one of its goals the hiding of its location. It may be a very long time before a rule-based TSA system is capable of accepting and using conceptual structures having causal links of this kind, but an eventual operational system must have such a capability.

A major advantage of combining some of the NLP operations with TSA operations in a production system is that the NLP processes can share the production system's extensive database. For example, the properties (class, type, category, etc.) and most recent tracks of the platforms are needed by both the NLP and the TSA reasoning operations. The organization of a production system shared in this manner is discussed further in the next two sections.

3.2 Integration of NLP with STAMMER2

Figure 3-1 shows the major parts of a system which uses STAMMER2 to perform some of the NLP reasoning in addition to TSA reasoning. The files containing preprocessing functions, first-stage NLP functions and NLP oracles (LISP functions which would perform pattern matching and various other manipulations) would be loaded along with STAMMER2 files. Care would be needed initially to avoid giving a NLP function or variable the same name as a STAMMER2 function or variable. The rule-set would contain both NLP rules and TSA rules. The "message receipt" mechanism of STAMMER2 would require a small modification to permit the receipt of messages from multiple sources during a single run.

Temporary data used in the processing of the partly formatted messages would be stored as property lists and as variable bindings to word lists, numbers, etc. (The equivalent temporary data exist for TSA operations in STAMMER2 -- consider the data to be represented in the support boxes.) The temporary data would be used or operated on by the oracles. Permanent and semi-permanent data would be stored as assertions in STAMMER2's primary database, both for NLP and TSA. When an oracle is called (as an evaluation of a rule condition), its arguments generally are retrieved from this primary database. An example of an oracle which would be needed is one that compares the capabilities of candidate platforms against a reported activity when a sentence about the activity does not fully identify the platform. Another example of a needed oracle is one that performs consistency checks with position data.

```
                          2ND-STAGE NLP AND TSA IN STAMMER2
                                          ^
                        __________________ _________________
                       /                                    \

                                   * * * * * * * * *
                                   *               *
                      ---------------------- -------------
                      |2ND-STAGE   /  TSA  | |  RULE     |
                      |NLP RULES  /  RULES | |INTERPRETER|
     1ST-STAGE NLP    ---------------------- -------------
          ^               *            *      | *
 _________ __________
/                    \    *            *      | *
                       --------------------   | * -------------
                       |  NLP     / TSA   |   | * | OTHER     |
                       |ORACLES  / ORACLES|   | * | COMPUTING |
                       --------------------   | * |  SUPPORT  |
                          *            *      | * |(GEOMETRY, |
                          *            *      | * |CONFIDENCE,|
-----------     ---------------- -----------  | * |    ETC.)  |
|1ST-STAGE|*****|TEMPORARY DATA| |ASSERTION|<-| * -------------
|  NLP    |     | BASE USED BY | | DATABASE|*****
|         |---->| BOTH STAGES  |  ----------
-----------     ----------------       ^       ----------------
                  ^           ^        |       |USER SUPPORT  |
                  |           |     ---------  |(EXPLANATION, |
                  |           |     |MESSAGE|  |GRAPHICS, ETC)|
                  |           |     |RECEIPT|  ----------------
                  |  FORMATTED|DATA ---------
         NARRATIVE|  PERTINENT|TO        ^ ^
             LINES|  1ST-STAGE|NLP       | |__OTHER
                  |           |          |    SOURCES
                ----------------         |
PARTLY FORMATTED |   INITIAL   |FORMATTED|DATA FOR
---------------->|RESTRUCTURING|----------
    MESSAGES     |& EXTRACTION |2ND-STAGE NLP & TSA
                ----------------
```

Figure 3-1: Natural language processing shared by STAMMER2. The boxes represent components of a candidate experimental configuration.

The current version of STAMMER2 does not permit rules to be selectively activated or deactivated. If a partitioning of rules is needed for efficient operation, a modification enabling this should be attempted. Rule partitioning is a capability of Rosie, an alternative production system discussed in the next section.

3.3 Integration of NLP with Rosie

Rosie (A Rule-Oriented System for Implementing Expertise) is a development of The Rand Corporation. The most noticeable difference between Rosie and STAMMER2 systems is in the rule syntax -- Rosie rules are written in an English-like syntax while STAMMER rules are coded in statements involving assertions or oracles.

In an experimental configuration of a system using Rosie, a first-stage Interlisp program would restructure the message lines, extract pertinent formatted data, parse the narrative lines and then would create a temporary file containing data to be operated on in Rosie. As in the proposed system involving STAMMER2, the temporary data would include, for ambiguous conceptual structures, the candidate actions, actors, objects, etc., and their respective weights. Under a separate job, a ruleset in Rosie would read the file and store the data in Rosie syntax in the database, and NLP rulesets also activated for the purpose would perform further analyses. (Alternatively, some of the parsing could be performed with NLP rulesets employing Rosie's pattern-matching features.) Most of the manipulations performed via oracles in STAMMER2 would be performed in Rosie via "system rulesets," rulesets that access Lisp functions. As with the STAMMER2 version and its database of assertions, TSA and second-stage NLP activities would share a large database. Rosie syntax permits data to be stored in statements equivalent to STAMMER2 assertions and also in many other more complex syntax structures.

3.4 NLP in a Dedicated Production System

Little has been said, to this point, about the structure of the initial NLP unit in the experimental system configurations proposed in Sections 3.2 and 3.3. A rule-based approach would use packets of pattern-action rules similar to those constituting the grammar of Marcus' model of a parser [13]. Implementation of grammar rules would be awkward in STAMMER2 and Rosie and would best be accomplished in a system designed for and dedicated to NLP. Implementation of the "second-stage" NLP rules in this same system would be desirable, although access to the database of the

TSA system would be required. Implementation of all of the NLP rules in a single production system would permit expectation-based parsing using knowledge derived from the TSA database.

3.5 Conclusions

In addition to parsing, the computer understanding of tactical narrative from partly formatted messages will require reasoning processes which can establish cause and effect relationships between events, infer other events and activities, resolve ambiguities, etc. The implementation of frames and frame systems for situations of interest, including routine sequences of events, will permit the representation of some of the causal relationships, but a method has not yet been proposed to use these conceptual structures in a knowledge-based system performing TSA (tactical situation assessment). An adequate method of interpreting narrative data will require not only the facts from the message's formatted lines but also knowledge from an extensive database which includes properties, track histories and activities of platforms. The need for NLP functions to access the continually updated TSA database was a major consideration in the formulation of the system approaches outlined in this report.

Two systems proposed, one employing STAMMER2 and the other Rosie, would use linguistic knowledge in an initial NLP unit and domain knowledge in a production system already performing TSA. A better, but more difficult, approach would be to design a rule-based system suitable for both aspects -- syntactic and semantic/environmental -- of this special kind of NLP and to give the system access to the database maintained by a separate TSA system. A greater degree of expectation-based parsing can be achieved with this approach, since interaction is possible between the semantic/environmental processes and the syntactic processes.

In any near-term system, relatively few of the situations discussed in message narrative will be pertinent to the TSA rules implemented and, therefore, the number of situations that must be understood will be small. However, when the actor and/or object is not specified in a sentence concerning one of these situations, a complete understanding of that situation will sometimes require an understanding of a situation discussed in an earlier sentence. A system having a very high success rate would, therefore, require many times the number of frames of a system having a moderate success rate.

NOSC is involved in another program devised to develop a system for processing narrative tactical data. The system would query the message preparer to remove ambiguities before transmission. Although the system would not have access to the much greater collection of data at the data fusion site, it could remove many of the syntactic ambiguities concerning the action, actor and object, a problem discussed in Section 3.1. Data preprocessed by such a system would require significantly less processing by the data fusion system.

Studies of optimal system configurations will continue in conjunction with the work to extend the original NLP program.

4. DATABASE UPDATING

4.1 Introduction

This section discusses techniques for updating the database of the automated data fusion system described in Section 1.

4.2 Events

The term "event" will be used in a very general sense here, to refer to an activity, a determination of platform location, a signal intercept, or even a static situation. Some typical kinds of events are: refueling, search, sonobuoy laying, communication, radar detection, harassment, attack, damage assessment, and low fuel supply.

An event which is reported to have occurred or maybe to have occurred will be called an "actual" event. Those events which are only speculated or suspected to have occurred will have low confidence values associated with them. (There may be several different interpretations of an event, each with a confidence value, and these cases must be handled a special way in the database.)

A "virtual" event will refer to an event which has not occurred (at the time of the report) and may be a planned or requested activity, a prediction of enemy activity, etc.

Some virtual and actual events may be "supporting" events, e.g., the launching of aircraft may be in support of an attack of an enemy ship.

Examples of Narrative Reports of Virtual Events

Planned/Requested/Ordered Events

- Will strike Kobchik with airborne aircraft when hostilities commence.
- Have requested E-2 coverage 1800-2000T to ensure continuous locating data on Vietnamese units.
- Intend make sweep of area with active sonar.

- o Peterson directed to prosecute Victor with LAMPS and HS.

- o Pegasus currently enroute to attack Zhevny with gun. [The event concerning Pegasus being enroute is a supporting active event.]

- o Intend use eight strike aircraft from 0130T launch from SSSC in effort to locate two Kyndas. [If the 0130T launch has not yet occurred, it becomes a supporting virtual event to the virtual event(s) of locating the Kyndas.]

Predicted/Expected Events

- o Increased reconnaissance expected within 48 hours.

- o Increased air surveillance activity can be expected first light 7 Nov.

- o Anticipate approximately 60 aircraft are available for impending operations.

4.3 Event Records

The following is an outline of a procedure for creating and retiring event records.

a. For each event, whether an actual or virtual event, enter into the primary memory an event record of the appropriate kind for that event and link that record, as needed, with associated event records. (A "link" is formed by entering an assertion indicating the relationship and having as nodes or name-elements the two event-record labels.)

b. If another report of the same actual event occurs, update or correct the event record, as needed, and record the additional information source in that record. If the new report is not sufficiently correlated with a similar report, enter the new report into the primary memory and link the two with an assertion indicating the potential redundancy relationship.

c. If the event is an actual event, and a sufficiently similar virtual-event report is in the primary memory, then

(1) Link the actual-event record to the virtual-event record with an assertion which relates their respective labels and indicates the predictive relationship.

(2) If the virtual event is related to other events, e.g., a supporting relationship, connect the respective linking assertions instead to the actual event.

(3) Store the virtual-event record in the history file;

(4) Delete the virtual-event record from the primary memory.

d. If an active-event record has remained in the primary memory over a certain period of time (e.g., 1-2 days, depending on the event type), then

(1) Store the event record in the history file;

(2) In the primary memory, replace the event record with an event summary -- a minimum sized structure serving to indicate the nature of the event so its record can be retrieved from the history file if needed.

After a longer period of time,

(3) Delete the event summary from the primary memory.

e. If a virtual-event record has remained in the primary memory well past its expected time of occurrence, then

(1) Store the event record in the history file;

(2) Delete the event summary from the primary memory (optionally leaving for a time an assertion indicating its existence).

4.4 Experimental Results

Experiments were performed in Rosie for three kinds of events: aircraft launches, sightings and attacks. Rulesets were written for

- o Comparing actual-event reports with virtual-event reports and other actual-event reports
- o Combining actual-event reports found to be sufficiently similar
- o Retiring event reports into a history file.

These three functions are implemented, respectively, in the files "W-Update," "W-combine" and "W-Copy," listed in the appendix. The appendix begins with a typescript of an example involving these files and three others listed in the appendix -- WMsg1, WMsg2 and WMsg3, which enter messages in Rosie syntax, each message describing several events. The second message describes virtual events which predict the actual events described in the third message. The third message was entered another time (as Message #4) to see if reports of the same event would be correctly correlated. The messages represent the result of the initial processing of formatted and/or narrative data, discussed in earlier sections. The acts of starting the updating process (steps 13, 35 and 39 in the typescript) and the copying process (steps 45 and 46) are shown as human actions in the typescript, but can be performed easily by a "master" ruleset that decides when an action is appropriate.

The messages used in the example assume that aircraft launches support sightings and attacks. We also could have assumed that sightings support attacks. This decision usually would be made at an earlier stage, though -- one occurring during the message-understanding process. One of the useful functions of support relationships between events is that of providing lower bounds to times when event times are not given. A major future use is in the initial processing of the raw message; e.g., the NLP subsystems (see Figure 1-1) will have access to the data and may use them in cause and effect reasoning and in creating expectations.

The criteria used for comparing event times and other attributes and for assigning scores to the matching of attributes were somewhat arbitrary and highly oversimplified and should be

specified much more carefully in a real application. For example, when assigning a score for equivalence between two attributes, the case where they are identically equal should receive a higher score. Also, a variety of other factors would need to be considered, such as weapons, sensors and ranges. The prediction-match and redundancy-match scores were printed out only for evaluation of the program and such information would probably be optional for an operator. Other experiments, not shown, involved changing the attributes of events to varying degrees and observing the resulting scores.

In the initial experiments, the representation of events permitted multiple actors and multiple objects or victims, since sometimes co-located targets are sighted and attacked or two platforms participate in an attack. (The ruleset "Function count for attribute of event" in the file "W-Update" was involved in these experiments and is no longer needed.) Allowing multiple actors and objects/victims created serious difficulties, both in comparing event records and in retiring only the fulfilled part of virtual events. It was concluded that event records should have only one actor (which can be a "group" in the case of carrier aircraft, for example, where individual aircraft are not specified) and only one object or victim. When the object or victim is specified in a message as a composite and individual targets are reported in a later version, the initial event report can be split into more specific event records. A similar splitting can be made when a group named as the actor is later more fully defined, with the possible exception of aircraft. Additional rulesets will be needed for recognizing the need for splitting and accomplishing it. Also, event records involving simultaneous cooperative actions should be linked together with assertions indicating the relationship.

Experiments with a history file used the multiple database feature of Rosie, while an actual system would store such data offline. The ruleset (in the file "W-Copy" of the appendix) used in these experiments may accomplish the storing process in an inefficient manner, but it was written when the multiple database feature was first implemented and no documentation was available.

Rule 3 of "Procedure Update for New Event" in the file "W-Update" relates sightings to tracks in a form compatible with other programs in Rosie (tactical situation assessment rules and PTAPS rules). A round-about redundancy is created in that the sighted platform will be the object of the sighting and will also have a track which, in turn, has that sighting as one of its reports. However, the convenience of being able to run the

updating rulesets with other rulesets should compensate for the extra assertion. Step 44 illustrates the result of this rule for "TRACK #1."

4.5 Conclusions

It was observed from many examples of tactical narrative messages that the events described could be put in either of two categories: (1) those that occurred or were thought to have occurred and (2) those that were planned, ordered or predicted. These two categories of events are here termed "actual" events and "virtual" events, respectively.

Data concerning a virtual event are pertinent to the tactical situation picture until the predicted or planned event actually occurs (unless it is cancelled), at which time an event record of the actual event should replace the event record of the virtual event. To do this, first a method is needed to determine that the actual event was indeed "predicted" by the virtual event. Also, a method is needed to identify redundant reports and to update earlier event records, since the same event often is reported several times at different levels of reporting and rarely in an identical manner.

As a beginning step in developing these methods, simple versions of event records were created for three kinds of events: aircraft launches, sightings and attacks, and experimental programs were written to compare event records, to combine redundant actual-event records and to retire into a "history file" the event records which were no longer needed.

Early experiments indicated that to facilitate the updating and history-recording processes, event records should not have more than one actor or more than one object or victim, even at the expense of using several related event records to represent one event having several participants. (Whether or not to allow multiple sensors, multiple weapons, etc., has not yet been addressed.) This single actor or object/victim can be the name or label of a group, however, when the composition of the group is unimportant or unknown. In the latter case, if the group is later specified, usually it may be better to split the event report into one for each "member" of the group.

No attempt has been made yet to compare and combine redundant virtual-event reports, although this probably should be done.

Redundancy of virtual-report events will be less of a problem, since virtual events are not reported at as many levels of reporting as actual events and since their event reports generally will be retired to the history file much sooner than actual-event reports. The process of comparing virtual-event records will be more difficult than comparing actual-event records, though, since successive versions of a planned event may change.

The question of when to retire event records into the history file needs additional attention. One approach would be to calculate for each record, upon its creation or reorganization, a date-time of retirement, and then periodically sweep out all records whose times have expired. Some exceptions would have to be made, however; e.g., when an old sighting was the last sighting of a platform, its record should be kept much longer.

There is a variety of other problems that also need investigating. For example, an event record may be found to be a redundant version of two other event records which are contradictory to each other. A repetition of an event can be a problem; e.g., two attacks involving the same actor and victim may occur at times which are relatively close and not both specified precisely. In this case, clues such as "second" or "another" from the narrative are important. Further experiments will no doubt reveal many other problems.

5. SUMMARY AND RECOMMENDATIONS

In past efforts under this project, we have modelled an automated data fusion system and have experimented with its component processes, particularly natural language processing (NLP) and rule-based higher-order logic. In the recent work described in this report, we have centered our attention on the database of the system.

Knowledge about recent events, conceptually represented in a continually updated database, can be of considerable value in understanding incoming narrative. It can provide expectations, causes and enabling conditions useful in evaluating candidate interpretations of narrative data. The database also contains long term data, such as assertions about platform attributes and capabilities, which the NLP subsystem requires. For these reasons, access is highly desirable to the system's database by the NLP subsystems. Also, it appears that a rule-based approach should be taken in using the data to guide or control the processing of the narrative. Several ways of doing this were outlined in Section 3.

Other investigations in FY81 concerned the updating of the database of the data fusion system. An experimental program, written in Rosie (Rule-Oriented System for Implementing Expertise, a development of The Rand Corporation), compares event reports for redundancies, combines reports of the same event and retires old data into a history file in a form useful for event reconstruction. Section 4 discusses the experimental results and conclusions.

If the NLP controlling subsystems, the database updating subsystem and the final fusion processes are all to be rule-based processes which share the same database, as it appears they should be, they will require the same kind of rule interpreting mechanism. Furthermore, these processes ideally would be implemented in the same rule-based system, to facilitate coordination and interaction.

In conjunction with continuing investigations in the areas discussed in this report, several related areas should receive attention. One is the organization of the history file. A study of the potential uses of the history file and the kinds of information needed for each should influence the design of the database updating processes which create the history file. Another area is the employment of event records in the database

to understand new events and link their records with those of other events in a chain of events. For example, a report of an attack should create an expectation of a report of damage and permit a relational linking of the attack record with the damage record in a script representation. In some cases the reasoning needed to relate events can be used at the NLP stage and in other cases the same reasoning is needed at the database updating stage. Still another potential area of investigation is the use of event records to automatically modify status data concerning supplies and capabilities. Reports of missiles used in attacks could be used to update the missile inventory counts kept for both friendly and hostile platforms. (The detection of redundant reports is especially important in this case.) A report of a damaged system should alter the list of operational systems on a platform. The updating of status records in such situations is another possible application of a rule-based system.

The areas of needed investigation described in this report and also other important ones not discussed will require research efforts many times greater than can be supported under this research project. We encourage other researchers working in artificial intelligence and natural language processing to consider these data fusion problems in their design of new systems and techniques.

REFERENCES

1. New Methodologies for Automated Data Fusion, R.A. Dillard, Naval Ocean Systems Center, Technical Report 364, September 1978.

2. Higher Order Logic for Platform Identification in a Production System, R.A. Dillard, Naval Ocean Systems Center, Technical Document 288, October 17, 1979.

3. Experimental Tests of PTAPS Performance in Three Types of Production System Structures, R.A. Dillard, Naval Ocean Systems Center, Technical Document 385, September 17, 1980.

4. A Platform-Track Association Production Subsystem, R.A. Dillard, In Proceedings of the Fourth MIT/ONR Workshop on Command and Control. Held in San Diego, CA, June 1981.

5. Natural Language Processing Applied to Navy Tactical Messages, D.M. Keirsey, Naval Ocean Systems Center, Technical Document 324, February 1, 1980.

6. Text Understanding Techniques Applied to Partly Formatted Navy Tactical Messages, R.A. Dillard, Naval Ocean Systems Center, Technical Document 405, December 17, 1980.

7. STAMMER2: A Production System for Tactical Situation Assessment, D.C. McCall, P.H. Morris, D.F. Kibler and R.J. Bechtel, Naval Ocean Systems Center, Technical Document 298, Volumes 1 and 2, October 1979.

8. Object-Oriented Natural Language Understanding, D.M. Keirsey, Hughes Research Laboratories, Malibu, CA (Contract N66001-81-C-0086), Technical Report, August 1981.

9. A Framework for Representing Knowledge, M. Minsky, In The Psychology of Computer Vision, P.H. Winston (editor), pages 211-277. McGraw-Hill, 1975.

10. Scripts, Plans, Goals, and Understanding: An inquiry into human knowledge structures, R. Schank and R. Abelson, Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1977.

11. Knowledge Representation for Automated Understanding of Natural Language Discourse, G. Silva and C.A. Montgomery, In Computers and the Humanities, Vol. 11, pages 223-234, Pergamon Press, 1978.

12. STAMMER: System for tactical assessment of multisource messages, even radar, R.J. Bechtel and P.H. Morris, Naval Ocean Systems Center, Technical Document 252, Naval Ocean Systems Center, May 1979.

13. A Theory of Syntactic Recognition for Natural Language, M. Marcus, MIT Press, 1980.

APPENDIX: DATABASE UPDATING EXPERIMENT

CONTENTS

```
                 TM
       [ Rosie   Monday, August 24, 1981  2:07pm ]

<2> load w-update.
@procedure UPDATE for NEW-EVENT
@predicate EVENT1 does describe_an_event_of_the_type of EVENT2
@function PREDICTION-MATCH of VIRTUAL-EVENT with ACTUAL-EVENT
@function REDUNDANCY-MATCH of EVENT1 with EVENT2
@predicate THING1 is an equivalent of THING2
@function MAXIMUM of TWO-NUMBERS
@function ABSOLUTE-VALUE of NUMBERC
@function COUNT for ATTRIBUTE of EVENT
@function TIME_DIFFERENCE of TIMES
<3> load w-combine.
@procedure COMBINE EVENT1 with EVENT2
@function BETTER_DESCRIBED of TWO-PLATFORMS
@procedure RESOLVE ACFT-GROUP1 with ACFT-GROUP2
<4> load w-copy.
@procedure COPY RETIRED-EVENT into HISTORY-FILE
<5> load wmsg1.

Data entered from MESSAGE #1
<6> message #1?
[ MESSAGE #1 ]
    MESSAGE #1 did report AIRCRAFT-LAUNCH #1.
    MESSAGE #1 did report SIGHTING #1.
    MESSAGE #1 did report ATTACK #1.
    MESSAGE #1 is a message.

    "Aircraft from 220630 launch located and attacked Varyag at 220845"
    is a narrative of MESSAGE #1.
    PLATFORM #1 is a source of MESSAGE #1.

<7> Display the name of platform #1.
CONSTELLATION
<8> aircraft-launch #1?
[ AIRCRAFT-LAUNCH #1 ]
    AIRCRAFT-LAUNCH #1 did support SIGHTING #1.
    AIRCRAFT-LAUNCH #1 did support ATTACK #1.
    MESSAGE #1 did report AIRCRAFT-LAUNCH #1.
    AIRCRAFT-LAUNCH #1 is an aircraft-launch.
    AIRCRAFT-LAUNCH #1 is an event.
    220630 is a time of AIRCRAFT-LAUNCH #1.
    AIRCRAFT-GROUP #1 is an aircraft of AIRCRAFT-LAUNCH #1.
    `CONSTELLATION did launch AIRCRAFT at 220630' is a description
    of AIRCRAFT-LAUNCH #1.
    AIRCRAFT-LAUNCH #1 is actual.
```

```
<9> aircraft-group #1?
[ AIRCRAFT-GROUP #1 ]
    AIRCRAFT-GROUP #1 is an aircraft-group.
    PLATFORM #1 is a base of AIRCRAFT-GROUP #1.
    AIRCRAFT-GROUP #1 is an aircraft of AIRCRAFT-LAUNCH #1.
    AIRCRAFT-GROUP #1 is an actor of SIGHTING #1.
    AIRCRAFT-GROUP #1 is an actor of ATTACK #1.

<10> sighting #1?
[ SIGHTING #1 ]
    AIRCRAFT-LAUNCH #1 did support SIGHTING #1.
    MESSAGE #1 did report SIGHTING #1.
    SIGHTING #1 is an event.
    220845 is a time of SIGHTING #1.
    `VARYAG was sighted at 220845 by AIRCRAFT from CONSTELLATION'
    is a description of SIGHTING #1.
    SIGHTING #1 is a sighting.
    35.4 is a latitude of SIGHTING #1.
    -15.12 is a longitude of SIGHTING #1.
    VISUAL is a sensor of SIGHTING #1.
    AIRCRAFT-GROUP #1 is an actor of SIGHTING #1.
    PLATFORM #2 is an object of SIGHTING #1.
    SIGHTING #1 is actual.

<11> platform #2?
[ PLATFORM #2 ]
    PLATFORM #2 is a platform.
    VARYAG is a name of PLATFORM #2.
    DESTROYER is a type of PLATFORM #2.
    UR is a flag of PLATFORM #2.
    HOSTILE is an id of PLATFORM #2.
    KYNDA is a class of PLATFORM #2.
    PLATFORM #2 is an object of SIGHTING #1.
    PLATFORM #2 is a victim of ATTACK #1.

<12> attack #1?
[ ATTACK #1 ]
    AIRCRAFT-LAUNCH #1 did support ATTACK #1.
    MESSAGE #1 did report ATTACK #1.
    ATTACK #1 is an event.
    220845 is a time of ATTACK #1.
    `VARYAG was attacked at 220845 by AIRCRAFT from CONSTELLATION'
    is a description of ATTACK #1.
    AIRCRAFT-GROUP #1 is an actor of ATTACK #1.
    ATTACK #1 is an attack.
    PLATFORM #2 is a victim of ATTACK #1.
    ATTACK #1 is actual.

<13> For each event, go update for that event.
<14> sighting #1?
[ SIGHTING #1 ]
    AIRCRAFT-LAUNCH #1 did support SIGHTING #1.
    MESSAGE #1 did report SIGHTING #1.
    SIGHTING #1 is an event.
```

220845 is a time of SIGHTING #1.
`VARYAG was sighted at 220845 by AIRCRAFT from CONSTELLATION'
is a description of SIGHTING #1.
SIGHTING #1 is a sighting.
35.4 is a latitude of SIGHTING #1.
-15.12 is a longitude of SIGHTING #1.
VISUAL is a sensor of SIGHTING #1.
AIRCRAFT-GROUP #1 is an actor of SIGHTING #1.
PLATFORM #2 is an object of SIGHTING #1.
SIGHTING #1 is a report of TRACK #1.
SIGHTING #1 is actual.

<15> track #1?
[TRACK #1]
TRACK #1 is a track.
PLATFORM #2 is a platform of TRACK #1.
TRACK #1 is a track of PLATFORM #2.
SIGHTING #1 is a report of TRACK #1.

<16> load wmsg2.

Data entered from MESSAGE #2
<17> message #2?
[MESSAGE #2]
MESSAGE #2 did report AIRCRAFT-LAUNCH #2.
MESSAGE #2 did report SIGHTING #2.
MESSAGE #2 did report ATTACK #2.
MESSAGE #2 did report ATTACK #3.
MESSAGE #2 is a message.

"Intend launch additional acft at 221120T to locate and attack second Kynda (Admiral Golovko) and re-attack Varyag" is a narrative of MESSAGE #2.
PLATFORM #1 is a source of MESSAGE #2.

<18> aircraft-launch #2?
[AIRCRAFT-LAUNCH #2]
MESSAGE #2 did report AIRCRAFT-LAUNCH #2.
AIRCRAFT-LAUNCH #2 will support SIGHTING #2.
AIRCRAFT-LAUNCH #2 will support ATTACK #2.
AIRCRAFT-LAUNCH #2 will support ATTACK #3.
AIRCRAFT-LAUNCH #2 is an aircraft-launch.
AIRCRAFT-LAUNCH #2 is an event.
AIRCRAFT-GROUP #2 is an aircraft of AIRCRAFT-LAUNCH #2.
`CONSTELLATION will launch AIRCRAFT at 221120' is a description of AIRCRAFT-LAUNCH #2.
221120 is a planned-time of AIRCRAFT-LAUNCH #2.
AIRCRAFT-LAUNCH #2 is virtual.

<19> Display the name of (the base of aircraft-group #2).
CONSTELLATION
<20> sighting #2?
[SIGHTING #2]

```
    MESSAGE #2 did report SIGHTING #2.
    AIRCRAFT-LAUNCH #2 will support SIGHTING #2.
    SIGHTING #2 is an event.
    `ADMIRAL GOLOVKO will be located after 221120 by AIRCRAFT from
    CONSTELLATION' is a description of SIGHTING #2.
    SIGHTING #2 is a sighting.
    AIRCRAFT-GROUP #2 is an actor of SIGHTING #2.
    PLATFORM #3 is an object of SIGHTING #2.
    SIGHTING #2 is virtual.

<21> platform #3?
[ PLATFORM #3 ]
    PLATFORM #3 is a platform.
    ADMIRAL GOLOVKO is a name of PLATFORM #3.
    DESTROYER is a type of PLATFORM #3.
    UR is a flag of PLATFORM #3.
    HOSTILE is an id of PLATFORM #3.
    KYNDA is a class of PLATFORM #3.
    PLATFORM #3 is an object of SIGHTING #2.
    PLATFORM #3 is a victim of ATTACK #2.

<22> attack #2?
[ ATTACK #2 ]
    MESSAGE #2 did report ATTACK #2.
    AIRCRAFT-LAUNCH #2 will support ATTACK #2.
    ATTACK #2 is an event.
    `ADMIRAL GOLOVKO will be attacked after 221120 by AIRCRAFT from
    CONSTELLATION' is a description of ATTACK #2.
    AIRCRAFT-GROUP #2 is an actor of ATTACK #2.
    ATTACK #2 is an attack.
    PLATFORM #3 is a victim of ATTACK #2.
    ATTACK #2 is virtual.

<23> attack #3?
[ ATTACK #3 ]
    MESSAGE #2 did report ATTACK #3.
    AIRCRAFT-LAUNCH #2 will support ATTACK #3.
    ATTACK #3 is an event.
    `VARYAG will be attacked after 221120 by AIRCRAFT from
    CONSTELLATION' is a description of ATTACK #3.
    AIRCRAFT-GROUP #2 is an actor of ATTACK #3.
    ATTACK #3 is an attack.
    PLATFORM #2 is a victim of ATTACK #3.
    ATTACK #3 is virtual.

<24> load wmsg3.

Data entered from MESSAGE #3
<25> message #3?
[ MESSAGE #3 ]
    MESSAGE #3 did report AIRCRAFT-LAUNCH #3.
    MESSAGE #3 did report SIGHTING #3.
    MESSAGE #3 did report SIGHTING #4.
    MESSAGE #3 did report ATTACK #5.
```

```
    MESSAGE #3 did report ATTACK #4.
    MESSAGE #3 is a message.

"Acft fm 22120 launch attacked Varyag at 221200. Located Adm Golovko
and conducted strike at 221245" is a narrative of MESSAGE #3.
    PLATFORM #1 is a source of MESSAGE #3.

<26> aircraft-launch #3?
[ AIRCRAFT-LAUNCH #3 ]
    AIRCRAFT-LAUNCH #3 did support SIGHTING #3.
    AIRCRAFT-LAUNCH #3 did support ATTACK #4.
    AIRCRAFT-LAUNCH #3 did support SIGHTING #4.
    AIRCRAFT-LAUNCH #3 did support ATTACK #5.
    MESSAGE #3 did report AIRCRAFT-LAUNCH #3.
    AIRCRAFT-LAUNCH #3 is an aircraft-launch.
    AIRCRAFT-LAUNCH #3 is an event.
    221120 is a time of AIRCRAFT-LAUNCH #3.
    AIRCRAFT-GROUP #3 is an aircraft of AIRCRAFT-LAUNCH #3.
    `CONSTELLATION did launch AIRCRAFT at 221120' is a description
    of AIRCRAFT-LAUNCH #3.
    AIRCRAFT-LAUNCH #3 is actual.

<27> Display the name of (the base of (the aircraft of aircraft-launch #3)).
CONSTELLATION
<28> sighting #3?
[ SIGHTING #3 ]
    AIRCRAFT-LAUNCH #3 did support SIGHTING #3.
    MESSAGE #3 did report SIGHTING #3.
    SIGHTING #3 is an event.
    221200 is a time of SIGHTING #3.
    `VARYAG was located at 221200 by AIRCRAFT from CONSTELLATION'
    is a description of SIGHTING #3.
    SIGHTING #3 is a sighting.
    35.85 is a latitude of SIGHTING #3.
    -16.61 is a longitude of SIGHTING #3.
    VISUAL is a sensor of SIGHTING #3.
    AIRCRAFT-GROUP #3 is an actor of SIGHTING #3.
    PLATFORM #2 is an object of SIGHTING #3.
    SIGHTING #3 is actual.

<29> attack #4?
[ ATTACK #4 ]
    AIRCRAFT-LAUNCH #3 did support ATTACK #4.
    MESSAGE #3 did report ATTACK #4.
    ATTACK #4 is an event.
    221200 is a time of ATTACK #4.
    `VARYAG was attacked at 221200 by AIRCRAFT from CONSTELLATION'
    is a description of ATTACK #4.
    AIRCRAFT-GROUP #3 is an actor of ATTACK #4.
    ATTACK #4 is an attack.
    PLATFORM #2 is a victim of ATTACK #4.
    ATTACK #4 is actual.
```

```
<30> sighting #4?
[ SIGHTING #4 ]
    AIRCRAFT-LAUNCH #3 did support SIGHTING #4.
    MESSAGE #3 did report SIGHTING #4.
    SIGHTING #4 is an event.
    221245 is a time of SIGHTING #4.
    `ADMIRAL GOLOVKO was located at 221245 by AIRCRAFT from
    CONSTELLATION' is a description of SIGHTING #4.
    SIGHTING #4 is a sighting.
    35.9 is a latitude of SIGHTING #4.
    -16.72 is a longitude of SIGHTING #4.
    VISUAL is a sensor of SIGHTING #4.
    AIRCRAFT-GROUP #3 is an actor of SIGHTING #4.
    PLATFORM #3 is an object of SIGHTING #4.
    SIGHTING #4 is actual.

<31> attack #5?
[ ATTACK #5 ]
    AIRCRAFT-LAUNCH #3 did support ATTACK #5.
    MESSAGE #3 did report ATTACK #5.
    ATTACK #5 is an event.
    221245 is a time of ATTACK #5.
    `ADMIRAL GOLOVKO was attacked at 221245 by AIRCRAFT from
    CONSTELLATION' is a description of ATTACK #5.
    AIRCRAFT-GROUP #3 is an actor of ATTACK #5.
    ATTACK #5 is an attack.
    PLATFORM #3 is a victim of ATTACK #5.
    ATTACK #5 is actual.

<32> display every virtual event.
AIRCRAFT-LAUNCH #2
SIGHTING #2
ATTACK #2
ATTACK #3
<33> display every actual sighting.
SIGHTING #1
SIGHTING #3
SIGHTING #4
<34> display every actual attack.
ATTACK #1
ATTACK #4
ATTACK #5
<35> for each event (e) which message #3 did report,
     display e and go update for e.
AIRCRAFT-LAUNCH #3

The prediction-match of AIRCRAFT-LAUNCH #3 with AIRCRAFT-LAUNCH #2 is 30


The redundancy-match of AIRCRAFT-LAUNCH #3 with AIRCRAFT-LAUNCH #1 is 0
SIGHTING #3
```

```
The prediction-match of SIGHTING #3 with SIGHTING #2 is 0

The redundancy-match of SIGHTING #3 with SIGHTING #1 is 0

The redundancy-match of SIGHTING #3 with SIGHTING #4 is 0
ATTACK #4

The prediction-match of ATTACK #4 with ATTACK #2 is 0

The prediction-match of ATTACK #4 with ATTACK #3 is 25

The redundancy-match of ATTACK #4 with ATTACK #1 is 0

The redundancy-match of ATTACK #4 with ATTACK #5 is 0
SIGHTING #4

The prediction-match of SIGHTING #4 with SIGHTING #2 is 25

The redundancy-match of SIGHTING #4 with SIGHTING #1 is 0

The redundancy-match of SIGHTING #4 with SIGHTING #3 is 0
ATTACK #5

The prediction-match of ATTACK #5 with ATTACK #2 is 25

The prediction-match of ATTACK #5 with ATTACK #3 is 0

The redundancy-match of ATTACK #5 with ATTACK #1 is 0

The redundancy-match of ATTACK #5 with ATTACK #4 is 0
<36> For each event (el), for each event (e2),
     if el did predict e2, display el and display e2.
AIRCRAFT-LAUNCH #2
AIRCRAFT-LAUNCH #3
SIGHTING #2
SIGHTING #4
ATTACK #2
ATTACK #5
ATTACK #3
ATTACK #4
<37> load wmsg3.

Data entered from MESSAGE #4
<38> message #4?
[ MESSAGE #4 ]
    MESSAGE #4 did report AIRCRAFT-LAUNCH #4.
    MESSAGE #4 did report SIGHTING #5.
    MESSAGE #4 did report SIGHTING #6.
    MESSAGE #4 did report ATTACK #7.
    MESSAGE #4 did report ATTACK #6.
    MESSAGE #4 is a message.


"Acft fm 22120 launch attacked Varyag at 221200. Located Adm Golovko
```

and conducted strike at 221245" is a narrative of MESSAGE #4.
PLATFORM #1 is a source of MESSAGE #4.

<39> For each event (e) which message #4 did report,
display e and go update for e.
AIRCRAFT-LAUNCH #4

The prediction-match of AIRCRAFT-LAUNCH #4 with AIRCRAFT-LAUNCH #2 is 30

The redundancy-match of AIRCRAFT-LAUNCH #4 with AIRCRAFT-LAUNCH #1 is 0

AIRCRAFT-LAUNCH #3 has been combined with AIRCRAFT-LAUNCH #4.

The redundancy-match of AIRCRAFT-LAUNCH #4 with AIRCRAFT-LAUNCH #3 is 25

SIGHTING #5

The prediction-match of SIGHTING #5 with SIGHTING #2 is 0

The redundancy-match of SIGHTING #5 with SIGHTING #1 is 0

SIGHTING #3 has been combined with SIGHTING #5.

The redundancy-match of SIGHTING #5 with SIGHTING #3 is 30

The redundancy-match of SIGHTING #5 with SIGHTING #4 is 0

The redundancy-match of SIGHTING #5 with SIGHTING #6 is 0
ATTACK #6

The prediction-match of ATTACK #6 with ATTACK #2 is 0

The prediction-match of ATTACK #6 with ATTACK #3 is 25

The redundancy-match of ATTACK #6 with ATTACK #1 is 0

ATTACK #4 has been combined with ATTACK #6.

The redundancy-match of ATTACK #6 with ATTACK #4 is 30

The redundancy-match of ATTACK #6 with ATTACK #5 is 0

The redundancy-match of ATTACK #6 with ATTACK #7 is 0
SIGHTING #6

The prediction-match of SIGHTING #6 with SIGHTING #2 is 25

The redundancy-match of SIGHTING #6 with SIGHTING #1 is 0

SIGHTING #4 has been combined with SIGHTING #6.

```
The redundancy-match of SIGHTING #6 with SIGHTING #4 is 30

The redundancy-match of SIGHTING #6 with SIGHTING #5 is 0
ATTACK #7

The prediction-match of ATTACK #7 with ATTACK #2 is 25

The prediction-match of ATTACK #7 with ATTACK #3 is 0

The redundancy-match of ATTACK #7 with ATTACK #1 is 0

ATTACK #5 has been combined with ATTACK #7.

The redundancy-match of ATTACK #7 with ATTACK #5 is 30

The redundancy-match of ATTACK #7 with ATTACK #6 is 0
<40> For each event (el), for each event (e2),
     if el was combined with e2, display el and display e2.
AIRCRAFT-LAUNCH #3
AIRCRAFT-LAUNCH #4
SIGHTING #3
SIGHTING #5
ATTACK #4
ATTACK #6
SIGHTING #4
SIGHTING #6
ATTACK #5
ATTACK #7
<41> aircraft-launch #3?
[ AIRCRAFT-LAUNCH #3 ]
    AIRCRAFT-LAUNCH #3 is an event.
    AIRCRAFT-LAUNCH #3 was combined with AIRCRAFT-LAUNCH #4.

<42> sighting #3?
[ SIGHTING #3 ]
    SIGHTING #3 is an event.
    SIGHTING #3 was combined with SIGHTING #5.

<43> sighting #5?
[ SIGHTING #5 ]
    AIRCRAFT-LAUNCH #4 did support SIGHTING #5.
    MESSAGE #4 did report SIGHTING #5.
    MESSAGE #3 did report SIGHTING #5.
    SIGHTING #5 is an event.
    221200 is a time of SIGHTING #5.
    `VARYAG was located at 221200 by AIRCRAFT from CONSTELLATION'
    is a description of SIGHTING #5.
    SIGHTING #5 is a sighting.
    35.85 is a latitude of SIGHTING #5.
    -16.61 is a longitude of SIGHTING #5.
    VISUAL is a sensor of SIGHTING #5.
    AIRCRAFT-GROUP #4 is an actor of SIGHTING #5.
    PLATFORM #2 is an object of SIGHTING #5.
    SIGHTING #5 is a report of TRACK #1.
```

```
    SIGHTING #3 was combined with SIGHTING #5.
    SIGHTING #5 is actual.

<44> track #1?
[ TRACK #1 ]
    TRACK #1 is a track.
    PLATFORM #2 is a platform of TRACK #1.
    TRACK #1 is a track of PLATFORM #2.
    SIGHTING #1 is a report of TRACK #1.
    SIGHTING #5 is a report of TRACK #1.

<45> For each event (e) which message #1 did report,
     display e and go copy e into history-file
     and forget about e and assert e is stored in history-file.
AIRCRAFT-LAUNCH #1
SIGHTING #1
ATTACK #1
<46> for each event (e) which message #2 did report,
     display e and go copy e into history-file
     and, forget about e and assert e is stored in history-file.
AIRCRAFT-LAUNCH #2
SIGHTING #2
ATTACK #2
ATTACK #3
<47> aircraft-launch #1?
[ AIRCRAFT-LAUNCH #1 ]
    AIRCRAFT-LAUNCH #1 is stored in HISTORY-FILE.

<48> attack #3?
[ ATTACK #3 ]
    ATTACK #3 is stored in HISTORY-FILE.

<49> history-file?
[ HISTORY-FILE ]
    AIRCRAFT-LAUNCH #1 is stored in HISTORY-FILE.
    SIGHTING #1 is stored in HISTORY-FILE.
    ATTACK #1 is stored in HISTORY-FILE.
    AIRCRAFT-LAUNCH #2 is stored in HISTORY-FILE.
    SIGHTING #2 is stored in HISTORY-FILE.
    ATTACK #2 is stored in HISTORY-FILE.
    ATTACK #3 is stored in HISTORY-FILE.

<50> display the databases.
<GLOBAL, HISTORY-FILE, PRIMARY-MEMORY>
<51> activate history-file.
<52> display every actual event.
AIRCRAFT-LAUNCH #1
SIGHTING #1
ATTACK #1
<53> dispay every virtual event.
DISPAY => DISPLAY ? yes
AIRCRAFT-LAUNCH #2
SIGHTING #2
ATTACK #2
```

```
ATTACK #3
<54> aircraft-launch #1?
[ AIRCRAFT-LAUNCH #1 ]
    AIRCRAFT-LAUNCH #1 did support SIGHTING #1.
    AIRCRAFT-LAUNCH #1 did support ATTACK #1.
    MESSAGE #1 did report AIRCRAFT-LAUNCH #1.
    AIRCRAFT-LAUNCH #1 is an event.
    `CONSTELLATION did launch AIRCRAFT at 220630' is a description
    of AIRCRAFT-LAUNCH #1.
    220630 is a time of AIRCRAFT-LAUNCH #1.
    AIRCRAFT-LAUNCH #1 is an aircraft-launch.
    AIRCRAFT-GROUP #1 is an aircraft of AIRCRAFT-LAUNCH #1.
    AIRCRAFT-LAUNCH #1 is actual.

<55> attack #3?
[ ATTACK #3 ]
    MESSAGE #2 did report ATTACK #3.
    ATTACK #3 did predict ATTACK #6.
    AIRCRAFT-LAUNCH #2 will support ATTACK #3.
    ATTACK #3 is an event.
    `VARYAG will be attacked after 221120 by AIRCRAFT from
    CONSTELLATION' is a description of ATTACK #3.
    AIRCRAFT-GROUP #2 is an actor of ATTACK #3.
    PLATFORM #2 is a victim of ATTACK #3.
    ATTACK #3 is an attack.
    ATTACK #3 is virtual.

<56> logout.
```

```
[: W-UPDATE  Created 24-Aug-81 1:50pm, edit by NOSC :]

Procedure Update for new-event.

Locals xmatch.

[1] If the new-event is actual,
for each virtual event (ve) which does
    describe_an_event_of_the_type of the new-event,
let the xmatch be
    the prediction-match of ve with the new-event
and (if the xmatch >= 25,
    assert ve did predict the new-event)
and (if the xmatch < 25 and the xmatch >= 20,
    assert ve did somewhat_predict the new-event)
and send {return,
"The prediction-match of ", the new-event, " with ", ve, " is ",
   the xmatch, return}.

[2] If the new-event is actual,
for each actual event (ae)
    which does describe_an_event_of_the_type of the new-event,
if ae ~= the new-event,
let the xmatch be
     the redundancy-match of ae with the new-event
and (if the xmatch >= 25,
    go combine ae with the new-event
    and forget about ae
    and assert ae is an event
    and ae was combined with the new-event)
and (if the xmatch < 25 and the xmatch >= 20,
assert the new-event does resemble ae)
and send {return,
"The redundancy-match of ", the new-event, " with ", ae, " is ",
   the xmatch, return}.

[3] If the new-event is an actual sighting
and there is an object (plat) of the new-event
and plat is a platform,
(if there is no track of plat,
 create a track whose platform is plat
 and which is a track of plat)
and assert the new-event is a report of (the track of plat).

[4] Deny the new-event is unprocessed.

End ruleset.
```

```
Predicate event1 does describe_an_event_of_the_type of event2.

[1] Match the event1 against
    {anything (bind type1 to the name), " #", anything}
and match the event2 against
    {anything (bind type2 to the name), " #", anything}
and (if type1 = type2
    conclude true).

[2] Conclude false.

End ruleset.


Function prediction-match of virtual-event with actual-event.

Locals mintime, score1, score2, score3, score4, scoretotal,
   mintime, xdelta.

[1] Let the score1 be 0 and the score2 be 0 and the score3 be 0
    and the score4 be 0 and the mintime be -1.

[2] If the virtual-event is an aircraft-launch
and there is an aircraft (vac) of the virtual-event
and there is an aircraft (aac) of the actual-event
and there is a base (vb) of vac
and there is a base (ab) of aac,
(if vb ~= ab, produce 0)
and (if vb = ab, let the score1 be 15).

[3] If there is a planned-time (pt) of the virtual-event
and there is a time (t) of the actual-event,
let the xdelta be the time_difference of <t, pt>
and (if the xdelta > 120 or the xdelta < -60, produce 0)
and (if the xdelta < 60 or the xdelta < -15, let the score2 be 5)
and (if the xdelta <= 30, let the score2 be 10)
and (if the xdelta = 0, let the score2 be 15).

[4] If there is no planned-time of the virtual-event
and there is a time of the actual-event,
for each event (se) which will support the virtual-event,
if (there is a planned-time (st) of se
   or there is a time (st) of se),
let the mintime be the maximum of <the mintime, st>.
```

```
[5] If the mintime > 0
and there is a time (t) of the actual-event,
let the xdelta be the time_difference of <t, the mintime>
and (if the xdelta < 0 or the xdelta > 240, produce 0)
and (if the xdelta < 180, let the score2 be 5)
and (if the xdelta < 90, let the score2 be 10).

[6] If there is an actor (actorl) of the virtual-event
and there is an actor (actor2) of the actual-event,
if actorl is an equivalent of actor2,
let the scorel be 5,
otherwise produce 0.

[7] If there is an object (objectl) of the virtual-event
and there is an object (object2) of the actual-event,
if objectl is an equivalent of object2,
let the score3 be 10,
otherwise produce 0.

[8] If there is a victim (victiml) of the virtual-event
and there is an victim (victim2) of the actual-event,
if victiml is an equivalent of victim2,
let the score3 be 10,
otherwise produce 0.

[9] Produce the scorel + the score2  + the score3 + the score4.

End ruleset.


Function redundancy-match of eventl with event2.

Locals scorel, score2, score3, score4, xdelta.

[1] Let the scorel be 0 and the score2 be 0 and the score3 be 0
    and the score4 be 0.

[2] If the eventl is an aircraft-launch
and there is an aircraft (acl) of the eventl
and there is an aircraft (ac2) of the event2
and there is a base (bl) of acl
and there is a base (b2) of ac2,
(if bl ~= b2, produce 0)
and (if bl = b2, let the scorel be 10).
```

```
[3] If there is a time (tl) of the eventl
and there is a time (t2) of the event2,
let the xdelta be the absolute-value
     of (the time_difference of <t2, tl>)
and (if the xdelta > 90, produce 0)
and (if the xdelta < 60,let the score2 be 5)
and (if the xdelta < 30, let the score2 be 10)
and (if the xdelta = 0, let the score2 be 15).

[4] If (there is no time of the eventl
   or there is no time of the event2),
for each event (el) which did support the eventl,
if there is an event (e2) such that
   (e2 did support the event2
    and e2 does describe_an_event_of_the_type of el),
(if el = e2 let the score2 be 10)
and (if el does resemble e2 and the score2 ~=10
    let the score2 be 5).

[5] If there is a latitude (latl) of eventl
and there is a latitude (lat2) of event2
and there is a longitude (lonl) of eventl
and there is a longitude (lon2) of event2,
if the absolute-value of (latl - lat2) < .01
and the absolute-value of (lonl - lon2) < .01,
   let the score2 be the score2 + 5,
otherwise produce 0.

[6] If there is an actor (actorl) of the eventl
and there is an actor (actor2) of the event2,
if actorl is an equivalent of actor2,
let the scorel be 5,
otherwise produce 0.

[7] If there is an object (objectl) of the eventl
and there is an object (object2) of the event2,
if objectl is an equivalent of object2,
let the score3 be 10,
otherwise produce 0.

[8] If there is a victim (victiml) of the eventl
and there is an victim (victim2) of the event2,
if victiml is an equivalent of victim2,
let the score3 be 10,
otherwise produce 0.

[9] Produce the score4 + the score3 + the score2 + the scorel.

End ruleset.
```

```
Predicate thing1 is an equivalent of thing2.

[1] if the thing1 = the thing2 conclude true.

[2] If (there is no name of the thing1
or there is no name of the thing2)
and there is a class (c1) of the thing1
and there is a class (c2) of the thing2
and c1 = c2
conclude true.

[3] If (there is no class of the thing1
or there is no class of the thing2)
and there is a type (t1) of the thing1
and there is a type (t2) of the thing2
and t1 = t2
conclude true.

[4] If the thing1 is an aircraft-group
and the thing2 is an aircraft-group
and the base of the thing1 = the base of the thing2
conclude true.

[5] Conclude false.

End ruleset.


Function maximum of two-numbers.

Locals number1, number2.

[1] Let the number1 be the member at 1 of the two-numbers.

[2] Let the number2 be the member at 2 of the two-numbers.

[3] If the number1 > the number2, produce the number1.

[4] Produce the number2.

End ruleset.


Function absolute-value of numberC.

[1] If the numberC < 0
  produce the negation of the numberC.

[2] Produce the numberC.

End ruleset.
```

```
Function count for attribute of event.

Locals count.

[1] Let the count be 0.

[2] If the attribute = actor,
for each actor of the event,
let the count be the count + 1.

[3] If the attribute = object,
for each object of the event,
let the count be the count + 1.

[4] If the attribute = victim,
for each victim of the event,
let the count be the count + 1.

[5] Produce the count.

End ruleset.


Function Time_Difference of times.

Locals atime, btime.

[1] Let the atime be the member at 1 of the times
and the btime be the member at 2 of the times.

[2] Match the atime against
{something (bind aday to the number),
2 nonblanks (bind ahour to the number),
2 nonblanks (bind aminute to the number)}

and match the btime against
{something (bind bday to the number),
2 nonblanks (bind bhour to the number),
2 nonblanks (bind bminute to the number)}

and (if aday - bday = 0
     produce aminute - bminute + 60 * (ahour - bhour))

and (if aday - bday ~= 0
     produce aminute - bminute
             + 60 * (ahour - bhour + 24 * (aday - bday))).

End ruleset.
```

```
[: W-COMBINE  Created 24-Aug-81 2:04pm, edit by NOSC :]

Procedure combine event1 with event2.

[1] If there is no time of the event2
and there is a time (t1) of the event1
let t1 be the time of the event2.

[2] If there is a time (t2) of the event2
and there is a time (t1) of the event1
and t1 ~= t2,
assert t1 is a different_reported_time of the event2.

[3] For each different_reported_time (t) of the event1,
assert t is a different_reported_time for the event2.

[4] If event2 is an aircraft-launch
and there is an aircraft (group1) of the event1,
if there is no aircraft of the event2,
let group1 be the aircraft of the event2,
otherwise go resolve a1
    with (the aircraft of the event2).

[5] If there is no actor of the event2
and there is an actor (a1) of the event1,
let a1 be the actor of the event2.

[6] If there is an actor (a1) of the event1
and there is an actor (a2) of the event2
and a1 ~= a2,
let (the better_described of <a1, a2>)
   be the actor of the event2.

[7] If there is an object (o1) of the event1
and there is an object (o2) of the event2
and o1 ~= o2,
let (the better_described of <a1, a2>)
   be the object of the event2.

[8] If there is no object of the event2
and there is an object (o1) of the event1,
let o1 be the object of the event2.

[9] If there is a victim (v1) of the event1
and there is a victim (v2) of the event2
and v1 ~= v2,
let v1 be the victim of the event2.
```

```
[10] If there is no victim of the event2
and there is a victim (v1) of the event1,
let v1 be the victim of the event2.

[11] If there is a latitude (lat1) of event1
and there is no latitude of event2,
let lat1 be the latitude of event2
and let (the longitude of event1) be the longitude of event2.

[12] For each message (m) which did report the event1,
assert m did report the event2.

[13] For each event (e) which event1 did support,
assert event2 did support e.

[14] For each event (e) which did support event1,
assert e did support event2.

[15] For each event (e) which event1 will support,
assert event2 will support e.

[16] Send {return, the event1, " has been combined with ",
           the event2, ".", return}.

End ruleset.


Function better_described of two-platforms.

Locals plat1, plat2.

[1] Let the plat2 be the member at 2 of the two-platforms.

[2] If there is a name of the plat2, produce the plat2.

[3] Let the plat1 be the member at 1 of the two-platforms.

[4] If there is a name of the plat1, produce the plat1.

[5] If there is a class of the plat2, produce the plat2.

[6] If there is a class of the plat1, produce the plat1.

[7] If there is a type of  he plat2, produce the plat2.

[8] If there is a type of the plat1, produce the plat1.

[9] Produce the plat2.

End ruleset.
```

Procedure resolve acft-group1 with acft-group2.

[1] If there is no base of the acft-group2
and there is a base (b) of the acft-group1,
let b be the base of the acft-group2.

[2] For each event whose actor is the acft-group1
let the acft-group2 be the actor of that event.

[3] Forget about the acft-group1.

End ruleset.

[: W-COPY Created 11-Aug-81 8:27am, edit by NOSC :]

Procedure copy retired-event into history-file.

Locals xevent, xtime, xacft, xbase.

[1] If the retired-event is an aircraft-launch
let the xevent be aircraft-launch.

[2] If the retired-event is a sighting
let the xevent be sighting.

[3] If the retired-event is an attack
let the xevent be attack.

[4] If the retired-event is virtual,

activate the history-file
and assert the retired-event is a virtual event
and activate primary-memory

and (If there is a planned-time (t) of the retired-event,
let the xtime be t,
otherwise let the xtime be unspecified)

and (For each event (pe) which the retired-event did predict,
activate the history-file
and assert the retired-event did predict pe
and activate primary-memory)

and (For each event (se) which will support the retired-event,
activate the history-file
and assert se will support the retired-event
and activate primary-memory).

[5] For each event (se) which the retired-event will support,
activate the history-file
and assert the retired-event will support se
and activate primary-memory.

[6] If the retired-event is actual,

activate the history-file
and assert the retired-event is an actual event
and activate primary-memory

and (If there is a time (t) of the retired-event,
let the xtime be t,
otherwise let the xtime be unspecified)

and (For each event (pe) which did predict the retired-event,
activate the history-file
and assert pe did predict the retired-event
and activate primary-memory)

and (For each event (se) which did support the retired-event,
activate the history-file
and assert se did support the retired-event
and activate primary-memory)

and (For each event (se) which the retired-event did support,
activate the history-file
and assert the retired-event did support se
and activate primary-memory).

[7] If there is an aircraft (ac) of the retired-event,
 let the xacft be ac and
  (if there is a base (b) of ac,
   let the xbase be b,
   otherwise let the xbase be unspecified),
otherwise let the xacft be unspecified.

[8] For each description (d) of the retired-event,
activate the history-file
and assert d is a description of the retired-event
and activate primary-memory.

[9] For each actor (ar) of the retired-event,
activate the history-file
and assert ar is an actor of the retired-event
and activate primary-memory.

[10] For each object (o) of the retired-event,
activate the history-file
and assert o is an object of the retired-event
and activate primary-memory.

[11] For each victim (v) of the retired-event,
activate the history-file
and assert v is a victim of the retired-event
and activate primary-memory.

[12] For each weapon (w) of the retired-event,
activate the history-file
and assert w is a weapon of the retired-event
and activate primary-memory.

[13] For each sensor (s) of the retired-event,
activate the history-file
and assert s is a sensor of the retired-event
and activate primary-memory.

[14] For each message (m) which did report the retired-event,
activate the history-file
and assert m did report the retired-event
and activate primary-memory.

[15] Activate the history-file.

[16] If the retired-event is virtual
and the xtime ~= unspecified,
assert the xtime is a planned-time of the retired-event.

[17] If the retired-event is actual
and the xtime ~= unspecified,
assert the xtime is a time of the retired-event.

[18] If the xevent = aircraft-launch,
assert the retired-event is an aircraft-launch
and let the aircraft of the retired-event be the xacft
and if the xbase ~= unspecified,
assert the xbase is a base of (the aircraft of the retired-event).

[19] If the xevent = sighting,
assert the retired-event is a sighting.

[20] If the xevent = attack,
assert the retired-event is an attack.

[21] Activate primary-memory.

End ruleset.

```
[: WMSG1  Created 19-Aug-81 2:52pm, edit by NOSC :]

[rule 1] activate primary-memory.

[rule 2] create a platform (p) whose name is Constellation
and let carrier be the type of p
and US be the flag of p
and friend be the ID of p.

[rule 3] create a platform whose name is Varyag
and whose class is Kynda.

[rule 4] create a platform whose name is Admiral Golovko
and whose class is Kynda.

[rule 5] for each platform (p) whose class is Kynda,
let destroyer be the type of p
and hostile be the ID of p
and UR be the flag of p.

[rule 6] create an aircraft-launch (al)
which is an actual event
and whose time is 220630
and create an aircraft-group (ag)
    whose base is the platform whose name is Constellation
and let ag be the aircraft of al
and assert
  `Constellation did launch aircraft at 220630'
     is a description of al

and  create a sighting (s)
which is an actual event
and whose time is 220845
and whose latitude is 35.40
and whose longitude is -15.12
and whose sensor is visual
and whose actor is ag
and whose object is the platform
    whose name is Varyag
and assert al did support s and
`Varyag was sighted at 220845 by aircraft from Constellation'
    is a description of s

and create an attack (atk)
which is an actual event
and whose time is 220845
and whose actor is ag
and whose victim is the platform
    whose name is Varyag
and assert al did support atk and
```

```
`Varyag was attacked at 220845 by aircraft from Constellation'
   is a description of atk

and create a message (msg)
whose narrative is
"Aircraft from 220630 launch located and attacked Varyag at 220845"
and whose source is the platform whose name is Constellation
and assert msg did report each of al, s and atk

and send {return,
"Data entered from ", msg, return}.
```

```
[: WMSG2  Created 16-Aug-81 11:39am, edit by NOSC :]

[rule 1] create an aircraft-launch (al)
which is a virtual event
and whose planned-time is 221120
and create an aircraft-group (ag)
whose base is the platform whose name is Constellation
and let ag be the aircraft of al
and assert
`Constellation will launch aircraft at 221120'
   is a description of al

and create a sighting (s)
which is a virtual event
and whose actor is ag
and whose object is the platform
   whose name is Admiral Golovko
and assert al will support s and
`Admiral Golovko will be located after 221120 by aircraft from
Constellation' is a description of s

and create an attack (atk1)
which is a virtual event
and whose actor is ag
and whose victim is the platform
   whose name is Admiral Golovko
and assert al will support atk1 and
`Admiral Golovko will be attacked after 221120 by aircraft from
Constellation' is a description of atk1

and create an attack (atk2)
which is a virtual event
and whose actor is ag
and whose victim is the platform
   whose name is Varyag
and assert al will support atk2 and
`Varyag will be attacked after 221120 by aircraft from Constellation'
   is a description of atk2

and create a message (msg)
whose narrative is
"Intend launch additional acft at 221120T to locate and attack second
Kynda (Admiral Golovko) and re-attack Varyag"
and whose source is the platform whose name is Constellation
and assert msg did report each of al, s, atk1 and atk2

and send {return,
"Data entered from ", msg, return}.
```

```
[: WMSG3  Created 24-Aug-81 9:16am, edit by NOSC :]

[rule 1] create an aircraft-launch (al)
which is a actual event
and whose time is 221120
and create an aircraft-group (ag)
whose base is the platform whose name is Constellation
and let ag be the aircraft of al
and assert
`Constellation did launch aircraft at 221120'
   is a description of al

and create a sighting (s1)
which is a actual event
and whose time is 221200
and whose latitude is 35.85
and whose longitude is -16.61
and whose sensor is visual
and whose actor is ag
and whose object is the platform
   whose name is Varyag
and assert al did support s1 and
`Varyag was located at 221200 by aircraft from
Constellation' is a description of s1

and create an attack (ak1)
which is a actual event
and whose time is 221200
and whose actor is ag
and whose victim is the platform
   whose name is Varyag
and assert al did support ak1 and
`Varyag was attacked at 221200 by aircraft from Constellation'
   is a description of ak1

and create a sighting (s2)
which is a actual event
and whose time is 221245
and whose latitude is 35.9
and whose longitude is -16.72
and whose sensor is visual
and whose actor is ag
and whose object is the platform
   whose name is Admiral Golovko
and assert al did support s2 and
`Admiral Golovko was located at 221245 by aircraft from
Constellation' is a description of s2
```

```
and create an attack (ak2)
which is a actual event
and whose time is 221245
and whose actor is ag
and whose victim is the platform
   whose name is Admiral Golovko
and assert al did support ak2 and
`Admiral Golovko was attacked at 221245 by aircraft from
Constellation' is a description of ak2

and create a message (msg)
whose narrative is
"Acft fm 22120 launch attacked Varyag at 221200. Located Adm Golovko
and conducted strike at 221245"
and whose source is the platform whose name is Constellation
and assert msg did report each of al, sl, s2, ak2 and akl

and send {return,
"Data entered from ", msg, return}.
```

ATE
LME